

AMENDMENTS TO THE CLAIMS

Please amend the claims as follows.

- ab  
DI
1. (Currently Amended) A method of converting text to speech comprising:  
receiving a list of textual units, where ~~each~~ said textual units ~~is one of a~~ in the list  
comprise words, a prefixes, or a and suffixes; and  
for each textual unit ~~comprising a word~~ in the list,  
locating an associated speech sample in a memory; and  
appending said associated speech sample to an output signal.
- C1
2. (Original) The method of claim 1 wherein one said textual unit in said list is  
indicated as not having an associated speech sample in memory and said method further  
comprises:  
passing said indicated textual unit to a secondary text to speech engine;  
receiving a speech sample converted from said indicated textual unit from said secondary  
text to speech engine; and  
appending said converted speech sample to said output signal.

3. (Original) The method of claim 2 wherein each said speech sample in said memory comprises a processed recording of a voice talent and said secondary text to speech engine comprises a phonetic text to speech engine based on said voice talent.

4. (Original) The method of claim 1 wherein a consecutive plurality of said textual units in said list represent a whole word, said method further comprising:

C/ for each textual unit in said consecutive plurality of said textual units, locating an associated speech sample in said memory;

creating a speech unit by splicing together said plurality of associated speech samples;  
and

appending said speech unit to said output signal.

5. (Original) The method of claim 4 further comprising, after said splicing, processing said speech unit to remove discontinuities.

6. (Original) A method of pre-processing a text file comprising:
- receiving a text file;
- parsing said text file into textual units, where each said parsed textual unit is one of a word, a prefix or a suffix; and
- for each one of said parsed textual units, if said one of said parsed textual units corresponds to a stored textual unit in a vocabulary of textual units, adding said stored textual unit to a list.
7. (Original) The method of claim 6 further comprising, for each one of said parsed textual units, if said one of said parsed textual units does not correspond to one of said stored textual units,
- marking said parsed textual unit as being out of vocabulary; and
- adding said marked textual unit to said list.
8. (Original) The method of claim 7 where said marking comprises pre-pending a character to said textual unit.

9. (Currently Amended) A text to speech converter comprising:

means for receiving a list of textual units, where ~~each~~ said textual units ~~is one of a~~ in the list comprise words, a prefixes, or a and suffixes; and

for each textual unit ~~comprising a word~~ in the list,

means for locating an associated speech sample in a memory; and

means for appending said associated speech sample to an output signal.

10. (Currently Amended) A text to speech converter comprising a processor operable to:

receive a list of textual units, where ~~each~~ said textual units ~~is one of a~~ in the list comprise words, a prefixes, or a and suffixes; and

for each textual unit ~~comprising a word~~ in the list,

locate an associated speech sample in a memory; and

append said associated speech sample to an output signal.

11. (Currently Amended) A computer readable medium for providing program control to a processor, said processor included in a text to speech converter, said computer readable medium adapting said processor to be operable to:

receive a list of textual units, where ~~each~~ said textual units ~~is one of a~~ in the list comprise words, a prefixes, ~~or a~~ and suffixes; and

for each textual unit ~~comprising a word~~ in the list,

locate an associated speech sample in a memory; and

append said associated speech sample to an output signal.

12. (Currently Amended) A text to speech conversion system comprising:

a text file pre-processor operable to:

receive a text file;

parse said text file into textual units, where each said parsed textual unit is one of a word, a prefix or a suffix; and

for each one of said parsed textual units, if said one of said parsed textual units corresponds to a stored textual unit in a vocabulary of textual units, add said stored textual units to a list;

and a textual unit processor operable to:

receive said list of textual units, where ~~each~~ said textual units ~~is one of a~~ in the list comprise words, a prefixes, or a and suffixes;

for each textual unit in said list ~~comprising a word~~:

locate an associated speech sample in a memory; and

append said associated speech sample to an output signal.

13. (Canceled).

14. (Currently Amended) A data structure including a field for a textual unit, and a field for a speech sample associated with said textual unit, a field for a frequency of a first portion of the speech sample that exceeds an amplitude threshold, and a field for a frequency of a last portion of the speech sample that exceeds the amplitude threshold, where said textual unit is one of a word, a prefix, or a suffix.

C/ 15. (Previously Presented) The data structure of claim 14 further comprising a field for a phoneme that said textual unit starts with, and a field for a phoneme that the textual unit ends with.

16. (Previously Presented) The method of claim 7 further comprising:  
passing said marked textual unit to a secondary text to speech engine;  
receiving a speech sample converted from said marked textual unit from said secondary text to speech engine; and  
appending said converted speech sample to said output signal.

17. (Previously Presented) The method of claim 8 further comprising:  
passing said marked textual unit to a secondary text to speech engine;  
receiving a speech sample converted from said marked textual unit from said secondary text to speech engine; and  
appending said converted speech sample to said output signal.

18. (Previously Presented) The text to speech conversion system of claim 12 wherein:

said pre-processor is further operable to:

for each one of said parsed textual units, if said one of said parsed textual units does not correspond to one of said stored textual units, marking said parsed textual unit as being out of vocabulary and add said marked textual unit to said list; and

said textual unit processor further comprises:

a secondary text to speech engine operable to receive said marked textual unit and convert said marked textual unit into a speech sample.

19. (New) The data structure of Claim 14, further comprising at least one field for identifying:

an offset from a beginning of the speech sample to each zero crossing that follows a positive valued major peak in the first portion of the speech sample; and

an offset from an end of the speech sample to each zero crossing that follows a positive valued major peak in the last portion of the speech sample.

20. (New) The data structure of Claim 19, further comprising at least one field for identifying one or more peak values associated with each zero crossing.